CSC309H1 F

Programming on the Web

Fall 2024 Syllabus

Course Meetings

CSC309H1 F

Section	Day & Time	Delivery Mode & Location
LEC0101	Tuesday, 5:00 PM - 8:00 PM	In Person: KP 108
LEC0201	Thursday, 5:00 PM - 8:00 PM	In Person: BA 1170
LEC2001	Thursday, 5:00 PM - 8:00 PM	In Person: BA 1170

Refer to ACORN for the most up-to-date information about the location of the course meetings.

Course Contacts

Course Website: https://www.cs.toronto.edu/~kianoosh/courses/csc309/

Instructor: Kianoosh Abbasi
Email: csc309-2024-09@cs.toronto.edu
Office Hours and Location: Tuesdays and Thursdays 7-8, after lecture ends. Also at other times by appointment
Additional Notes: Use this address for course-related inquiries (e.g., special considerations, administrative topics, etc). This address is monitored by more people and your requests can be addressed of more quickly. A ticket is created for each email, so chances of your requests being missed is much lower. In general, allow 1-2 working days for your tickets to be addressed.

Instructor: Kianoosh Abbasi
Email: kianoosh@cs.toronto.edu
Office Hours and Location: Tuesdays and Thursdays 7-8, after lecture ends. Also at other times by appointment
Additional Notes: Use this email for personal or private communication with the instructor. Make sure to include the course code in the subject line of the email.

Course Overview

An introduction to software development on the web. The course covers the development of programs that operate on the web and survey of technological alternatives, with emphasis on modern web development technologies. Concepts, including the internet and the web, static client content, dynamic client content, dynamically served content, data management, web development processes, deployment, and containerization are discussed. It features a comprehensive project for students to showcase their understanding over the course content.

Course Learning Outcomes

At the end of the course, you will:

- Understand the concept of web, servers, and clients
- Gain solid knowledge about the different components of a web app
- Learn how develop the back-end of a web app with Next.js and Prisma
- Learn how to develop the front-end of a web app with React
- Learn how to publish and deploy your app to the internet using Docker

Prerequisites: None
Corequisites: None
Exclusions: None
Recommended Preparation: None
Credit Value: None

Recommended co-requisite: Introduction to Databases (CSC343)

Marking Scheme

Assessment	Percent	Details	Due Date
Midterm	10%	There will be a midterm exam during week 5 of the lectures. The exam will be held in person and will be closed-book. However, you will be allowed to bring one letter-sized paper with notes on both sides. The exam is designed to test your understanding of the concepts, architectures, and practices discussed in the course. There will not be any questions about specific syntax, commands, function names, etc.	2024-10-01,2024-10-03

Assessment	Percent	Details	Due Date
Project Part 1	20%	The project simulates a real-world web application that you might be asked to develop as a freelance web developer. Even though the scope is small, it is designed to give a sense of what creating a real website and working as a full-stack developer would entail. For the project, you can form groups of 2 or 3 members. You can also do it alone, but it is not recommended as the workload is excessive for one person. You will deliver the project in two parts. In part 1, you will deliver the backend of the application.	2024-10-27
Project Part 2	30%	In part 2, you will develop the frontend, connect the backend to it, and deliver the full web application.	2024-11-22

Assessment	Percent	Details	Due Date
Weekly Exercises	10%	Every week, you will be given a handful of small problems related to the topics discussed at the lecture. The exercises are designed to help you understand the concepts better and gain practical experience. Due to the educational nature of the exercises, the auto-grader will be provided to you, so you can check your answers before submission. You should submit your solutions to Markus every Thursday by 3pm. Note that because of the number of exercises, their relatively low impact on your final grade, and the availability of the auto-grader, no late submissions, extensions, or remark requests will be accommodated. Your best 9 out of 10 submissions will count for grading. Keep in mind that the exercises are meant to give you some initial practice with the topics. Therefore, it is important to do them yourself and understand the solutions. It will be possible to satisfy the auto-grader without fully developing the solution. But grading is not the purpose of the exercises, and it might impact your ability to do the project or exams.	No Specific Date
In-Person Final Exam	30%		Final Exam Period

Late Assessment Submissions Policy

No late submission allowed in Weekly Exercises. For the project, late submissions will be accepted with a penalty of 10% per day, up to a maximum of three days late.

Course Schedule

Week	Description
Week 1 Date	Lecture 1: HTML and CSS Sep 3rd & 5th Web architecture, server/client model, request/response, HTTP, course intro, HTML, CSS
Week 2 Date	Lecture 2: JavaScript Sep 10th & 12st Syntax, objects, functions, scope, arrow functions, DOM, elements
Week 3 Date	Lecture 3: Node.js, Next.js, and APIs Sep 17th & 19th Frontend/backend, Modern web apps, server-side JS, node, npm, Next.js, API, Rest framework
Week 4 Date	Lecture 4: Async, Models, and ORM Sep 24th & 26th Async programming, promises, async/await, MVC, Models, ORM, Prisma
Week 5 Date	Lecture 5: **[Midterm]** CRUD Oct 1st & 3rd **(different venue: TBD)** Prisma queries, Next.js API handlers and CRUD, validation
Week 6 Date	Lecture 6: Auth and Migration Oct 8th & 10th Auth, sessions, tokens, access and refresh, migrations
Week 7 Date	Lecture 7: React Oct 15th & 17th React intro, JSX, props, events, state
Week 8 Date	Lecture 8: Monorepo and Hooks Oct 22nd & 24th React in Next.js, monorepo, hooks, fetch API, auth and CORS
Week 9 Date	Lecture 9: TypeScript and Advanced CSS Nov 5th & 7th Context, Type safety, TypeScript, TailwindCSS, responsive design
Week 10 Date	Lecture 10: Deployment Nov 12th & 14th Dev vs prod, build, process manager, web server, DevOps

Week 11 Date	Lecture 11: Docker Nov 19th & 21st Docker intro and history, containers, images, volumes, Docker hub
Week 12 Date	Lecture 12: Optional topics Nov 26th & 28th TBD

Policies & Statements

About the course

An introduction to software development on the web. Concepts underlying the development of programs that operate on the web; survey of technological alternatives; greater depth on some technologies. Operational concepts of the internet and the web, static client content, dynamic client content, dynamically served content, n-tiered architectures, web development processes, and security on the web. Assignments involve increasingly more complex web-based programs.

Learning outcomes

At the end of the course, you will: Understand the concept of web, servers, and clients Gain solid knowledge about various components of a website (back-end and front-end) Learn how to create a static website with HTML and CSS Learn how develop the back-end of a website with Django Learn how to develop the front-end of a website with React

Late/Missed Assignments

No late submissions will be accepted for the weekly exercises. For the project, late submissions will be accepted with a penalty of 10% per day, up to a maximum of three days late.

If you are registered with Accessibility Services, you might be entitled to extensions on the project deadlines. In that case, email the instructor (via the course-related email) at least one week before the deadline and request the extension. Note that individual entitlements will be adjusted to the size of the team. For example, if one member of a team of two is entitled to 7 days of extensions, the team will be granted ⌈7/2⌉=4 days.

Students with Disabilities or Accommodation Requirements

Students with diverse learning styles and needs are welcome in this course. If you have an acute or ongoing disability issue or accommodation need, you should register with Accessibility Services (AS) at the beginning of the academic year by visiting https://studentlife.utoronto.ca/department/accessibility-services/. Without registration, you will not be able to verify your situation with your instructors, and instructors will not be advised about

your accommodation needs. AS will assess your situation, develop an accommodation plan with you, and support you in requesting accommodation for your course work. Remember that the process of accommodation is private: AS will not share details of your needs or condition with any instructor, and your instructors will not reveal that you are registered with AS.

Academic Integrity

Honesty and fairness are fundamental to the University of Toronto's mission. Plagiarism is a form of academic fraud and is treated very seriously. You are also expected to read the handout How Not to Plagiarize and to be familiar with the Code of Behavior on Academic Matters.

You are allowed to search the internet, use online resources, open-source codes, and generative AI (e.g., ChatGPT) to help you with the project. However, sharing even a small piece of code with other teams is strictly prohibited (whether giving or receiving). All work that was not written by you or your teammates must be cited in the code, including open-source codes and code created by generative AI.

Specific Medical Circumstances

To request special consideration, bring supporting documentation to the instructor in person during office hours at least one week in advance of the due date.

In the event of illness or another unforeseen event, please contact the instructor (via the course-related email) and provide supporting documentation within one week of the missed work. You must also declare your absence on Acorn. Do not wait until the due date has passed. It is always easier to make alternate arrangements before the due date.